Arbitrary Ratio Sample Rate Conversion Using B-Spline Interpolation for Software Defined Radio

Xiaojing Huang
University of Wollongong, Australia
huang@uow.edu.au

Abstract

Arbitrary ratio sampling rate conversion (SRC) structure using B-spline interpolation is proposed for software defined radio (SDR) in this paper. By combining SRC with SDR's transmitter/receiver filter, the constraint on SRC reconstruction filter can be relaxed, and an overall computational reduction can be achieved. The mixed-width B-spline is introduced so that both anti-imaging and anti-aliasing requirements for SRC are satisfied. The passband droop introduced by the B-spline interpolation is compensated by a linear phase digital filter incorporated in the SRC structure so that the overall frequency response approaches the desired frequency response of the SDR's transmitter/receiver filter. To make the proposed SRC structure applicable in practice, the mixed-width B-spline is further converted into uni-width B-spline, and the simplified implementation of the uni-width B-spline interpolation is also derived. A design example of the linear phase digital filter for the proposed SRC structure is given for an IEEE 802.11g wireless local area network (WLAN) SDR receiver, and the overall SRC complexity is analyzed.

1. Introduction

With the advance in digital signal processing and digital communication techniques, software defined radio (SDR) becomes a reality. For an SDR with multi-protocol and/or multi-band capabilities, sample rate conversion (SRC) is an essential element in the software architecture of the SDR [1]. Different sample rates are even used in a single communication protocol, such as the IEEE 802.11g wireless local area network (WLAN) specification [14], and thus SRC is a must if a single master clock implementation is required. As a kind of signal decimation and interpolation technique, a variety of SRC structures have been proposed over the past decades [2-13,17-18].

The most popular and computationally efficient approach for SRC is to use the cascaded integrator-comb (CIC) filter [2] which allows for a direct digital-digital SRC, i.e., the conversion is performed digitally with digital input and digital output but without an analog stage as theoretically required. Although this nature represents an advantage in terms of computational efficiency provided that the intermediate sample rate is not too high, it is actually the cause of its drawbacks such as wide transition band, passband droop, limitation to rational conversion ratio, and possible high intermediate sample rate. Recall that SRC is a process of signal reconstruction and then re-sampling. The reconstruction is ideally realized by a Nyquist lowpass filter, which converts the digital signal to analog signal without distortion. Therefore, SRC should preferably take a digital-analog-digital approach. The problem is how to select an appropriate reconstruction filter since the ideal Nyquist filter is neither possible nor necessary in practice.

In this paper, a digital-analog-digital SRC approach using B-spline interpolation is proposed, which overcomes the drawbacks of the digital-digital SRC using CIC filter. By this approach, the reconstruction filter is first sampled at an intermediate sampling rate and interpolated to approximate the desired transmitter /receiver filter. The effect of interpolation on the reconstruction filter's spectrum is then compensated digitally by an incorporated digital filter following a simple design procedure. A mixed-width B-spline interpolation function is used so that both anti-imaging and anti-aliasing requirements for SRC are satisfied. This mixed-width B-spline is further converted to uni-width B-spline and a simple implementation algorithm is also developed.

The rest of this paper is organized as follows. In Section 2, the SRC structures combined with SDR receiver and transmitter filters respectively are proposed, and the mixed-width B-spline which satisfies both anti-imaging and anti-aliasing requirements is introduced. The B-spline with single gate width is decomposed into basic interpolation functions in Section 3, resulting in a simplified B-spline interpolation implementation. Section 4 describes the frequency sampling method for designing a linear phase digital filter after compensating the passband droop introduced by the B-spline interpolation, and an SRC example is given for an IEEE 802.11g receiver. Finally, conclusions are drawn in Section 5.

2. SRC Structure Using B-Spline Interpolation

Our proposed digital-analog-digital SRC approach combined with receiver filter for an SDR receiver includes a virtual analog step and a re-sampling step. First, the sampled input sequence $x(kT_1)$ with sampling period T_1 is passed through a filter $h(t)$ to give a reconstructed analog signal

$$y(t) = \sum_{k=-\infty}^{\infty} x(kT_1)h(t - kT_1). \quad (1)$$

Second, $y(t)$ is re-sampled with sampling period T_2, resulting in a sample rate converted sequence

$$y(mT_2) = \sum_{k=-\infty}^{\infty} x(kT_1)h(mT_2 - kT_1). \quad (2)$$

This time-varying digital filtering can be simplified to a time-invariant operation if $h(t)$ is approximated by a digital filter $\hat{h}(nT_3)$ with a virtual intermediate sampling period T_3, i.e.,

$$h(t) = \sum_{n=-\infty}^{\infty} \hat{h}(nT_3)r(t - nT_3), \quad (3)$$

where $r(t)$ is an interpolation filter for approximating $h(t)$.

In the frequency domain, (3) can be expressed as

$$H(f) = \hat{H}\left(e^{j2\pi f T_3}\right)R(f), \quad (4)$$

where $H(f)$, $\hat{H}\left(e^{j2\pi f T_3}\right)$ and $R(f)$ denote the Fourier transforms of $h(t)$, $\hat{h}(nT_3)$ and $r(t)$ respectively. Assuming that the bandwidth of the desired signal is $2B$, one period of $\hat{H}\left(e^{j2\pi f T_3}\right)$ is simply determined from (4) by

$$\hat{H}\left(e^{j2\pi f T_3}\right) = \frac{H(f)}{R(f)}, \qquad \text{for } |f| \le B. \quad (5)$$

In this way, $\hat{h}(nT_3)$ can be designed according to the desired receiver filter frequency response $H(f)$ after compensating possible passband droop introduced by $R(f)$. Then, from (1) and (4) we have

$$\begin{aligned} Y(f) &= X\left(e^{j2\pi f T_1}\right)H(f) \\ &= \left\{X\left(e^{j2\pi f T_1}\right)R(f)\right\} \cdot \hat{H}\left(e^{j2\pi f T_3}\right), \end{aligned} \quad (6)$$

where $Y(f)$ and $X\left(e^{j2\pi f T_1}\right)$ are the Fourier transforms of $y(t)$ and $x(kT_1)$ respectively. Since the periodic nature of both $X\left(e^{j2\pi f T_1}\right)$ and $\hat{H}\left(e^{j2\pi f T_3}\right)$ produces image components and the re-sampling of $y(t)$ will cause spectrum aliasing in the desired frequency band, $R(f)$ must be chosen so that both anti-imaging and anti-aliasing requirements are met.

A suitable $R(f)$ which satisfies above criterion has been proposed in [17] as the trapezoidal interpolation filter. In general, $R(f)$ can be expressed as

$$R(f) = \left(T_1 \frac{\sin \pi f T_1}{\pi f T_1} e^{-j\pi f T_1}\right)^L \cdot \left(T_3 \frac{\sin \pi f T_3}{\pi f T_3} e^{-j\pi f T_3}\right)^M, \quad (7)$$

which is the product of $L + M$ sinc spectra. As shown in Fig. 1, the sinc spectrum $T_1 \frac{\sin \pi f T_1}{\pi f T_1} e^{-j\pi f T_1}$ has a natural imaging rejection ability for $X\left(e^{j2\pi f T_1}\right)$ whereas $T_3 \frac{\sin \pi f T_3}{\pi f T_3} e^{-j\pi f T_3}$ naturally rejects the images of $\hat{H}\left(e^{j2\pi f T_3}\right)$. The combined spectrum rolls off in the order of $f^{-(L+M)}$, so that it also offers superb anti-aliasing capability. When $L = M = 1$, (7) becomes the original trapezoidal interpolation filter.

In the time domain, $T_1 \frac{\sin \pi f T_1}{\pi f T_1} e^{-j\pi f T_1}$ corresponds to a rectangular pulse (i.e., gate) of width T_1, denoted by $g_{T_1}(t)$, and $T_3 \frac{\sin \pi f T_3}{\pi f T_3} e^{-j\pi f T_3}$ corresponds to a gate of width T_3, denoted by $g_{T_3}(t)$. Therefore,

$$r(t) = \underbrace{g_{T_1}(t) * \cdots * g_{T_1}(t)}_{L} * \underbrace{g_{T_3}(t) * \cdots * g_{T_3}(t)}_{M} \quad (8)$$

where $*$ denotes convolution, is a cascade of L gates of width T_1 and M gates of width T_3. We see that $r(t)$ is in fact a causal B-spline [15,16] but with mixed widths T_1 and T_3. Thus, when $r(t)$ is expressed as (8), (3) represents a mixed-width B-spline interpolation.

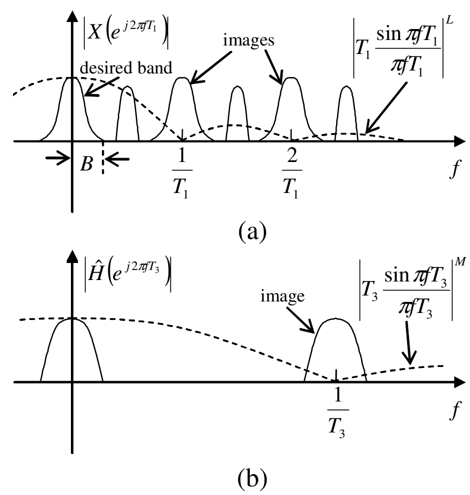

Fig. 1. Interpolation filter selection. (a) L-fold sinc spectrum with imaging rejection for $X\left(e^{j2\pi f T_1}\right)$. (b) M-fold sinc spectrum with imaging rejection for $\hat{H}\left(e^{j2\pi f T_3}\right)$.

By further choosing T_3 as an integer fraction of T_2, i.e.,

$$T_3 = \frac{T_2}{N}, \quad (9)$$

the SRC with receiver filter using B-spline interpolation for an SDR receiver is structured as follows. The received signal samples $x(kT_1)$ is first interpolated by $r(t)$ and re-sampled with sampling period T_3. The re-sampled sequence is then filtered by the time-invariant digital filter $\hat{h}(nT_3)$. The sample rate converted sequence $y(mT_2)$ is produced after down-sampling the filtered sequence by N.

Obviously, since the B-spline $r(t)$ incorporates gates with two different widths T_1 and T_3, direct implementation of this interpolation will be complicated. However, since $\sin x \cdot e^{-jx} = \frac{1}{2j}\left(1-e^{-j2x}\right)$, $R(f)$ can be rewritten as

$$R(f)=\left(\frac{1}{\pi f}\right)^{L+M}\left(\sin \pi f T_1 \cdot e^{-j\pi f T_1}\right)^{L}\left(\sin \pi f T_3 \cdot e^{-j\pi f T_3}\right)^{M}$$
$$=\left(\frac{1}{1-e^{-j2\pi f T_1}}\right)^{M}\cdot\left(T_1\frac{\sin \pi f T_1}{\pi f T_1}e^{-j\pi f T_1}\right)^{L+M}\cdot\left(1-e^{-j2\pi f T_3}\right)^{M}. \quad (10)$$

The frequency responses $\left(\frac{1}{1-e^{-j2\pi f T_1}}\right)^{M}$, $\left(T_1\frac{\sin \pi f T_1}{\pi f T_1}e^{-j\pi f T_1}\right)^{L+M}$ and $\left(1-e^{-j2\pi f T_3}\right)^{M}$ correspond respectively to a cascade of M integrators with system function $\left(\frac{1}{1-z^{-1}}\right)^{M}$ and sampling period T_1, a cascade of $L+M$ gates with a single width T_1, and a cascade of M combs with system function $\left(1-z^{-1}\right)^{M}$ and the sampling period T_3. The cascade of $L+M$ gates with the same width T_1 represents a uni-width B-spline

$$r_1(t)=\underbrace{g_{T_1}(t)*\cdots*g_{T_1}(t)}_{L+M}. \quad (11)$$

Therefore, the mixed-width B-spline interpolator $r(t)$ can be implemented by the uni-width B-spline interpolator $r_1(t)$ with preceding integrators and post combs after re-sampling. This SRC structure is shown in Fig. 2. The implementation of the uni-width B-spline interpolator $r_1(t)$ will be much easier than that of the original $r(t)$, and will be further simplified as described in the following section.

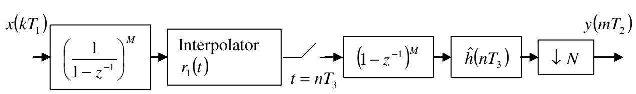

Fig. 2. Sample rate conversion using B-spline interpolation combined with receiver filter.

When the SDR works in transmitting mode, with the transmitter filter (or spectral shaping filter) being the same as $h(t)$ and using the same interpolation function $r(t)$, the combined SRC with spectral shaping can be also structured accordingly. First, $y(mT_2)$ is up-sampled by N and filtered by $\hat{h}(nT_3)$. Then, the sample rate converted transmitted signal $x(kT_1)$ is then produced after the mixed B-spline interpolation and re-sampling of the interpolated signal with sampling period T_1. When implementing $r(t)$ using the uni-width B-spline interpolator

$$r_3(t)=\underbrace{g_{T_3}(t)*\cdots*g_{T_3}(t)}_{L+M} \quad (12)$$

with the a single gate width T_3, this SRC structure is shown in Fig. 3.

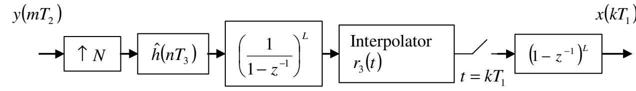

Fig. 3. Sample rate conversion using B-spline interpolation combined with transmitter filter.

Since an analog signal is virtually produced and re-sampled, the SRC structure using B-spline interpolation is able to cope with arbitrary conversion ratio. The incorporated digital filter not only compensates passband droop but also acts as receiver filter or spectral shaping filter, so that an overall computational reduction could be achieved. The design parameters L, M and N are selected according to the SRC anti-imaging and anti-aliasing requirements as well as the overall system complexity, as will be demonstrated by the design example described in Section 4.

3. Simplified B-Spline Implementation

In general, the nth order uni-width causal B-spline can be defined as [15,16]

$$\beta_T^{(n)}(t)=\underbrace{g_T(t)*\cdots*g_T(t)}_{n+1} \quad (13)$$

where

$$g_T(t)=\begin{cases}1, & 0\le t<T\\ 0, & \text{elsewhere}\end{cases} \quad (14)$$

is a gate of width T. Note that for the basic function defined in (14) the B-spline is causal instead of centered (non-causal). It is also of interest to know that this causal B-spline looks like the analog version of the digital CIC filter. By the above definition, the two B-splines $r_1(t)$ and $r_3(t)$ used in the SRC structures proposed in the previous section are simply $\beta_{T_1}^{(L+M-1)}(t)$ and $\beta_{T_3}^{(L+M-1)}(t)$. In this section, we will derive the simplified implementation of $\beta_T^{(n)}(t)$ with uni-width T. The result can be easily applied to $r_1(t)$ and $r_3(t)$, making the SRC structures applicable in practice.

As we all know, the B-spline is a piecewise continuous function of t with pulse duration $(n+1)T$. Each piece is an nth order polynomial in t with duration T. Suppose that the sampling period of the signal sequence to be interpolated is also T. If the B-spline interpolation is implemented directly, a set of $n+1$ samples of $\beta_T^{(n)}(t)$ must be calculated and convolved with input signal sequence to generate each interpolated output. Each sample of $\beta_T^{(n)}(t)$ is a sum of up to $n+1$ terms. Each term is a power of t to the power of up to n. We see that the direct implementation of the B-spline interpolation, like other polynomial interpolations, is hardly practical.

A centered B-spline can be efficiently implemented using the conventional Farrow structure [3,18]. For the causal B-spline introduced in this paper, an alternative simplified implementation can be realized by performing the B-spline interpolation recursively and decomposing the B-spline into a series of basic interpolation functions. To derive this simplified interpolation structure, let us firstly express $g_T(t)$ as the difference between a unit step $u(t)$ and its delayed version, i.e.,

$$g_T(t) = u(t) - u(t-T) = u(t) - D_T u(t) = (1 - D_T)u(t)\,, \quad (15)$$

where D_T denotes an operator which delays a function of t by T. Then, by denoting a set of normalized power function pulses of duration T as $\left(\frac{t}{T}\right)^m g_T(t)$, $m = 0,1,...,n$, we find out that the convolution of a lower order power function pulse with unit step can be decomposed as the sum (with a scale factor) of a higher order power function pulse plus a delayed unit step, i.e.,

$$\left(\frac{t}{T}\right)^{m-1} g_T(t) * u(t) = \int_0^t \left(\frac{\tau}{T}\right)^{m-1} g_T(\tau) d\tau = \frac{T}{m}\left(\frac{t}{T}\right)^m g_T(t) + \frac{T}{m} D_T u(t), \quad (16)$$

which is illustrated in Fig. 4.

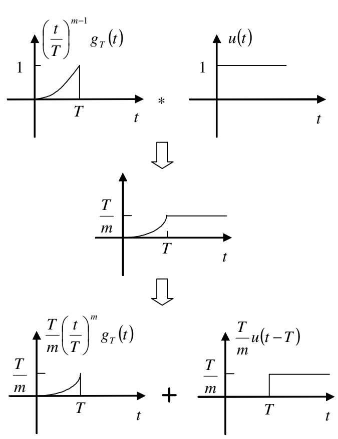

Fig. 4. The convolution of lower order power function pulse with unit step is decomposed as the sum of higher order power function pulse plus delayed unit step.

Applying the operator $1 - D_T$ to both sides of (16) and using (15), we have

$$\left(\frac{t}{T}\right)^{m-1} g_T(t) * g_T(t) = \frac{T}{m}\left[(1 - D_T)\left(\frac{t}{T}\right)^m g_T(t) + D_T g_T(t)\right]. \quad (17)$$

According to (13), the zero order B-spline is the gate $g_T(t)$, or, zero order power function pulse, i.e.,

$$\beta_T^{(0)}(t) = g_T(t). \quad (18)$$

According to (17) with $m = 1$, the first order B-spline is decomposed as

$$\beta_T^{(1)}(t) = g_T(t) * g_T(t) = T\left[(1 - D_T)\frac{t}{T} g_T(t) + D_T g_T(t)\right], \quad (19)$$

which is the first order power function pulse plus a zero order power function pulse with preceding operators $1 - D_T$ and D_T respectively. Using the result of the first order B-spline (19) and the formula (17), the second order B-spline is decomposed as

$$\beta_T^{(2)}(t) = \beta_T^{(1)}(t) * g_T(t) = T^2\left\{\left[D_T \cdot D_T + (1 - D_T)\cdot\frac{1}{2}\cdot D_T\right] g_T(t) + D_T\cdot(1 - D_T)\cdot\frac{t}{T} g_T(t) + (1 - D_T)\cdot\frac{1}{2}\cdot(1 - D_T)\cdot\left(\frac{t}{T}\right)^2 g_T(t)\right\}, \quad (20)$$

which is a sum of three power function pulses of orders from 0 to 2 with preceding composite operators $D_T \cdot D_T + (1 - D_T)\cdot\frac{1}{2}\cdot D_T$, $D_T \cdot (1 - D_T)$ and $(1 - D_T)\cdot\frac{1}{2}\cdot(1 - D_T)$ respectively.

Continuing the above recursion, the nth order B-spline can be decomposed into a sum of $n+1$ basic interpolation functions $\left(\frac{t}{T}\right)^m g_T(t)$, $m = 0,1,...,n$, preceded with respective composite operators. When this decomposed B-spline is used to interpolate an input discrete sequence with sampling period T, these composite operators can be applied to the discrete sequence first, resulting in a discrete delay and difference network with $n+1$ outputs followed by a bank of $n+1$ filters with these basic interpolation functions as their respective impulse responses. The flow graph of this simplified B-spline interpolation implementation is shown in Fig. 5, where the delay line z^{-1} corresponds to the operator D_T and the difference network $1 - z^{-1}$ corresponds to the operator $1 - D_T$. Since each output of the delay and difference network is a discrete sequence with sampling period T and the duration of the corresponding basic interpolation function is also T, the interpolated signal waveform is simply a successive concatenation of the corresponding basic interpolation function weighted by the respective input discrete samples without waveform overlapping among adjacent weighted basic interpolation functions. The calculation involved in the $(n+1)$-output network only requires divisions by integer numbers from 1 to n. For most SRC applications, the B-spline of order 2 already provides sufficient image rejection, so that there will be no multiplication in the network (dividing by 2 is equivalent to a right-shifting) and only 3 multiplications is needed in the filter bank (i.e., no multiplication for the zero order power function pulse, 1 multiplication for the first order power function pulse and 2 multiplications for the second order power function pulse).

4. Design Example

An example of SRC for the IEEE 802.11g WLAN receiver is given in this section. Since the IEEE 802.11g physical layer (PHY) packet has a single-carrier segment at 11 MHz sample rate and a multi-carrier segment at 20 MHz sample rate, an SRC component in the receiver is required to convert the sample rate from 20 MHz

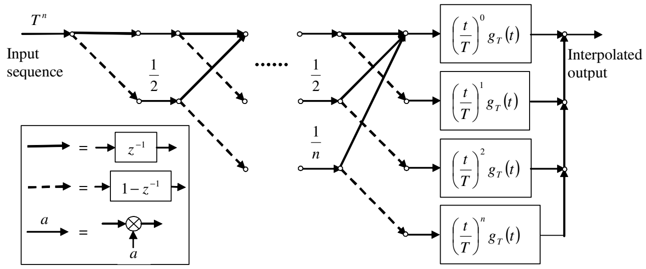

Fig. 5. Implementation of the nth order B-spline interpolation using an $(n+1)$-output delay and difference network and a bank of $n+1$ power function pulse filters.

($T_1 = \frac{1}{20}$ μS) to 11 MHz ($T_2 = \frac{1}{11}$ μS) for the single-carrier segment if we use only one master clock at 20 MHz. Obviously, if we use the CIC filter for this SRC, the intermediate frequency will be as high as 220 MHz, which is highly impractical.

For the single-carrier segment of the PHY packet, the impulse response of the transmitter filter is defined as a sinc function weighted by the continuous-time Hanning window [14], i.e.,

$$p(t) = f_W \frac{\sin(\pi f_W t)}{\pi f_W t} \cdot \frac{1}{2}\left[1+\cos\left(\frac{2\pi t}{T_{span}}\right)\right], \tag{21}$$

where $f_W = 16.25$ MHz is the bandwidth of the sinc function and $T_{span} = 0.8$ μS is the duration of $p(t)$. To reduce intersymbol interference (ISI), the receiver filter $h(t)$ should be chosen so that the overall frequency response of the transmitter filter and receiver filter has a smooth raised cosine spectrum $X_{rc}(f)$. Therefore, the frequency response of $h(t)$ is determined by $H(f) = \frac{X_{rc}(f)}{P(f)}$) where $P(f)$ is the Fourier transform of $p(t)$. Note that $h(t)$ is ideally a symmetric (zero-phase) function of an infinite duration.

After compensating the effect of B-spline interpolation, the digital filter $\hat{h}(nT_3)$ has a frequency response $\hat{H}(e^{j2\pi fT_3})$ with one period being $\frac{H(f)}{R(f)}$ as specified by (5). Since the B-spline interpolation introduces a time delay of $(LT_1 + MT_3)/2$ (see (7)), $\hat{h}(nT_3)$ shall be advanced by the same amount of time in order to have an overall zero-phase filter impulse response. However, in practice, it is more convenient to implement a linear phase finite impulse response (FIR) filter even though extra time delay will be introduced.

Suppose that 40 dB stopband attenuation of the receiver filter's frequency response is required. Fig. 6 shows the designed linear phase FIR filter $\hat{h}(nT_3)$ by the simple frequency sampling method ($\Delta f = \frac{1}{7T_2}$) using the second order B-spline ($L=1$, $M=2$, $L+M-1=2$) and the first order B-spline ($L=M=1, L+M-1=1$) respectively. Fig. 7 shows the corresponding actual receiver filter's magnitude frequency response $|H(f)|$, from which the image components are reduced by more than 45 dB using either the second order or the first order B-spline interpolation. However, for lower order B-spline interpolation, higher intermediate sample rate and longer FIR filter $\hat{h}(nT_3)$ have to be used in order to achieve similar anti-imaging and anti-aliasing performance. Consequently, computational load is actually heavier if lower order B-spling is used.

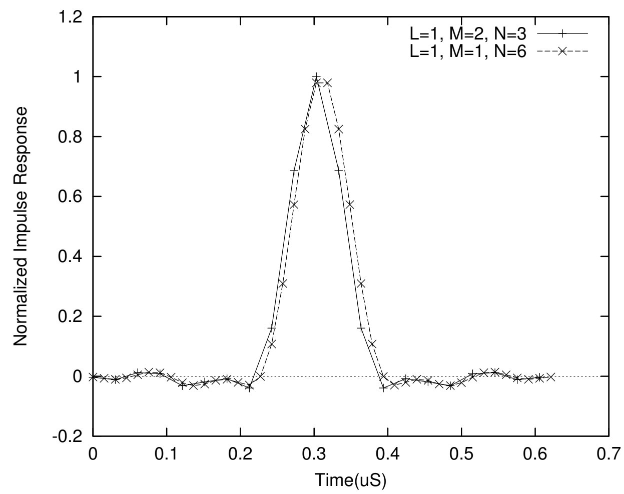

Fig. 6. The designed linear phase FIR filter $\hat{h}(nT_3)$ for the second and first order B-spline interpolations respectively ($X_{rc}(f)$ rolloff factor 0.8).

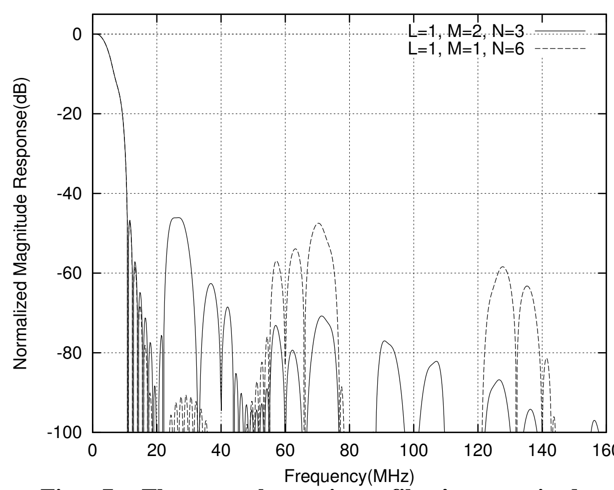

Fig. 7. The actual receiver filter's magnitude response $|H(f)|$ for the second and first order B-spline interpolations respectively ($X_{rc}(f)$ rolloff factor 0.8).

The complexity is calculated as follows. For the second order B-spline, the intermediate sample rate is chosen as $\frac{1}{T_3} = \frac{N}{T_2} = 33$ MHz (i.e., $N = 3$) and the length of $\hat{h}(nT_3)$ is $\frac{1}{\Delta f T_3} = 7N = 21$ (11 multiplications per sample are required for digital filtering after exploiting the linear phase property), whereas for the first order B-spline, the intermediate sample rate must be as high as $\frac{1}{T_3} = \frac{N}{T_2} = 66$ MHz (i.e., $N = 6$) and the length of $\hat{h}(nT_3)$ becomes $7N = 42$ (21 multiplications per sample for digital filtering). Therefore, with doubled intermediate sample rate and nearly doubled multiplications, the digital filtering in the first order B-spline case requires nearly four times computation than that in the second order B-spline case. We see that, even though the second order B-spline interpolation requires more computation (3 multiplications per sample) than the first order one (1 multiplication per sample), the overall complexity of SRC using the second order B-spline interpolation is actually greatly reduced.

5. Conclusions

As the analog counterpart of the digital CIC filter, the mixed-width B-spline interpolation has been introduced to meet both anti-imaging and anti-aliasing requirements for SRC, resulting in an arbitrary conversion ratio SRC structure combined with transmitter filter or receiver filter for SDR. The mixed-width B-spline is further converted into uni-width B-spline, and the implementation of the uni-width B-spline interpolation is simplified through decomposing the B-spline into a series of simple basic interpolation functions, resulting in a multiple output digital delay and difference network followed by a bank of short power function pulse filters. The passband droop introduced by the B-spline interpolation can be compensated by the digital filter incorporated in the SRC structures, and the simple frequency sampling method can be used for the linear phase digital filter design. By properly choosing the order of the B-spline and other design parameters such as intermediate sampling period and frequency sampling period, the anti-imaging and anti-aliasing requirements for SRC can be satisfied, and thus overall computational reduction can be achieved.

References

[1] T. Hentschel and G. Fettweis, “Sample rate conversion for software radio,” *IEEE Communications Magazine,* Vol. 38, No. 8, August 2000, pp. 142-150.

[2] E. B. Hogenauer, “An economical class of digital filters for decimation and interpolation,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. 29, No. 2, April 1981, pp. 155-162.

[3] C. W. Farrow, “A continuously variable digital delay element,” in *Proceedings of the 1988 IEEE Symposium on Circuits and Systems*, June 1988, Espoo, Finland, pp. 2641-2645.

[4] A. Y. Kwentus, Z. Jiang and A. N. Willson, Jr., “Application of filter sharpening to cascaded integrator-comb decimation filters,” *IEEE Transactions on Signal Processing,* Vol. 45, No. 2, February 1997, pp. 457-467.

[5] T.-C. Kuo, A. Y. Kwentus, and A. N. Willson, Jr., “A *programmable* interpolation filter for digital communications applications,” in *Proceedings of the 1998 IEEE International Symposium on Circuits and Systems*, Monterey, CA, June 1998. Vol. 2, pp. 97-100.

[6] L. *Wasserman* and A. N. Willson, Jr., “A variable-rate filtering system for digital communications,” in *Proceedings of the 1999* IEEE *International Conference on Acoustics, Speech and Signal Processing,* Vol. 3, pp. 1497-1500.

[7] H. J. Oh, S. Kim, G. Choi, and Y. H. Lee, “On the use of interpolated second-order polynomials for efficient filter *design* in programmable downconversion,” *IEEE Journal on Selected Areas in Communications*, Vol. 17, No. 4, April 1999, pp. 551-560.

[8] Y. Gao, L. Jia and H. Tenhunen, “A partial-polyphase VLSI *architecture* for very high speed CIC decimation filters,” in *Proceedings of the Twelfth Annual IEEE International ASIC/SOC Conference*, 1999, pp. 391-395.

[9] Y. Gao, L. Jia and H. Tenhunen, “An improved architecture and implementation of cascaded integrator-comb decimation filters,” in *Proceedings of the 1999 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, 1999, pp. 317-320.

[10] T. Hentschel and G. Fettweis, “Reduced complexity comb-filters for decimation and interpolation in mobile communications terminals,” *in Proceedings of the 6th IEEE International Conference on Electronics, Circuits and Systems*, Papfos, Cyprus, September 5-8, 1999, pp. 81-84.

[11] M. Henker, T. Hentschel and G. Fettweis, “Time-variant CIC-filters for sample rate conversion with arbitrary rational factors,” *in Proceedings of the 6th IEEE International* Conference *on Electronics, Circuits and Systems*, Papfos, Cyprus, September 5-8, 1999, pp. 67-70.

[12] A. Groth and H. G. Göckler, “Efficient minimum group delay block processing approach to fractional sample rate *conversion*,” in *Proceedings of the 2001 IEEE Symposium on Circuits and Systems,* Vol. 2, pp. 189-192.

[13] D. Babic, J. Vesma, and M. Renfors, “Decimation by irrational factor using CIC filter and linear interpolation,” in *Proceedings of the 2001 IEEE International Conference on Acoustics, Speech and Signal Processing,* Vol. 6, pp. 3677-3680.

[14] IEEE Standard 802.11g/D1.0, “Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: further higher-speed physical layer extension in the 2.4 GHz band,” November 2001.

[15] M. Unser, A. Aldroubi, and M. Eden, “B-spline signal processing: part I – theory,” *IEEE Transactions on Signal Processing*, Vol. 41, No. 2, February 1993, pp. 821- 833.

[16] M. Unser, “Splines: a perfect fit for signal and image processing,” *IEEE signal processing magazine*, Noverber 1999, pp. 22-38.

[17] X. Huang, Y. Li, and S. Nguyen, "Sample rate conversion by trapezoidal interpolation for software defined radio," in *Proceedings of PIMRC2003*, Vol. 1, pp. 135-139, Beijing, China, September 7-10, 2003.

[18] D. Babic and M. Renfors, “Power efficient structure for conversion between arbitrary sampling rates,” *IEEE Signal Processing Letters,* Vol. 12, No. 1, January 2005, pp. 1-4.